REMARKS

Reconsideration and withdrawal of the rejections of the application are respectfully requested in view of the foregoing amendments and following remarks.

I. STATUS OF THE CLAIMS AND FORMAL MATTERS

The Office Action indicates that claims 1-5 and 14-16 are pending in this application; however, the Office Action does not refer to dependent claim 26 that was newly presented in Applicant's amendment filed February 7, 2008 for entry and consideration on the merits. Applicant respectfully submits that claims 1-5, 14-16, and 26 are pending, and respectfully requests an indication that claim 26 has been entered and considered on the merits.

By this amendment, independent claims 1 and 14 are amended for additional clarity. No new matter has been added. It is submitted that the claims, as originally presented, were in full compliance with the requirements of 35 U.S.C. §112. Changes to claims are not made for the purpose of patentability within the meaning of 35 U.S.C. §101, §102, §103, or §112. Rather, these changes are primarily directed to form and are made simply for clarification.

II. REJECTIONS UNDER 35 USC § 103(a)

The Office Action maintains the rejection of claims 1-3, 5, and 14-16 under 35 USC § 103(a) as being unpatentable over *Computer Organization & Design The Hardware/Software Interface* to Patterson et al. (hereinafter, "Patterson") in view of *8051 Tutorial: Addressing Modes;* Vault Information Services, 2001, (hereinafter, "Vault"). The Office Action also maintains the rejection of claim 4 under 35 USC § 103(a) as being unpatentable over Patterson in view of Vault and common knowledge in the art, with the

Examiner taking Official Notice that the claim limitations stated in claim 4 were commonly known in the art at the time of the invention. Applicant respectfully traverses these rejections for at least the reasons present below.

As understood by Applicant, Patterson relates to a MIPS architecture having an ALU, a data memory, registers coupled to the ALU input, an instruction memory, and a program counter (PC).

As understood by Applicant, Vault relates to an 8051 instruction set that includes indirect addressing, allowing for extra bytes of Internal RAM found on an 8052 microcontroller to be accessed. In executing such an indirect addressing instruction, the 8051 analyzes the value of the R0 register, and loads the accumulator with the value held in the Internal RAM location addressed by the value of the R0 register.

Claim 1 recites, *inter alia*:

> An operation-processing device for performing operation processing based on an arbitrary operation program, said device comprising:
>
> ***a register array*** having a plurality of registers each for holding an arbitrary value based on a write address and a write control signal and for ***outputting the held value to a signal line based on a read address***;
>
> ***an operation portion having an input coupled to said signal line independent of an intervening addressable register such that a value read from said register array to said signal line based on a read address is capable of being provided to said input without further addressing a register, the operation portion being operable for performing an operation on said value read from said register array to said signal line;***
>
> an instruction-decoding portion for decoding an operation instruction from an operation program for operating said operation portion, ***wherein said operation instruction includes at least one bit indicative of an access method***; and

> an instruction-execution-controlling portion for controlling said register array and the operation portion in order to execute an operation instruction decoded by said instruction-decoding portion,
>
> ***wherein, in the event said at least one bit is indicative of a first access method, then said instruction-execution-controlling portion is operable for (i) selecting one of said registers based on said operation instruction, and (ii) based on a value held by said selected register, performing register-to-register addressing processing for selecting another of said registers of said register array; and***
>
> ***wherein, in the event said at least one bit is indicative of a second access method, then said instruction-execution-controlling portion is operable for selecting one of said registers based on said operation instruction, and not performing register-to-register addressing processing for selecting, based on a value held by said selected register, another of said registers of said register array.*** [Emphasis added.]

Applicant respectfully submits that Patterson and Vault, individually or in combination, do not teach or suggest the above combination of features recited in claim 1. More specifically, as Applicant previously maintained with respect to claim 1 as pending prior to the instant amendment, Patterson and Vault do not teach or suggest, *inter alia*, "a register array having a plurality of registers each for . . . *outputting the held value to a signal line based on a read address; an operation portion having an input coupled to said signal line* . . . [and] being operable for performing an operation on said value read from said register array to said signal line . . . and an instruction-execution-controlling portion . . . for selecting one of said registers . . . [and] based on a value held by said selected register . . . for performing register-to-register addressing processing for selecting another of said registers *of said register array*."

For example, in contrast to Applicant's claimed invention, neither Patterson nor Vault discloses or suggests register-to-register addressing within a register array that is coupled to an operation portion input as claimed (claim 1) at least insofar as (i) Patterson does not teach

or suggest that a value stored in a given one of the registers coupled to the ALU may be used to address another of the registers coupled to the ALU, and (ii) Vault merely discusses indirect addressing with respect to moving data *to a* register that outputs data to the ALU input *from a* storage location that cannot provide data to the ALU input without first providing the data to a register. Applicant has herein amended claim 1 to further clarify this distinction, namely, by further clarifying the architectural configuration among the register array in which register-to-register addressing is performed, the signal line to which a value held in the register array is output based on a read address, and an operation portion input coupled to the signal line: namely, claim 1 as amended (underlining indicating text added herein) recites, *inter alia,* "an operation portion having an input coupled to said signal line independent of an intervening addressable register such that a value read from said register array to said signal line based on a read address is capable of being provided to said input without further addressing a register."

Indeed, the Office Action acknowledges that "Patterson does not expressly disclose that based on a value held by said selected register, said instruction-execution-controlling portion performs register-to-register addressing processing for selecting another of said registers of said register array." The Office Action, however, alleges that it would have been obvious to provide Applicant's claimed invention (claim 1) by substituting the indirect addressing discussed in Vault for Patterson's direct addressing. More specifically, the Office action states the following:

> Vault teaches that based on a value held by said selected register (such as R0; see second page, Indirect Address section, first 5 paragraphs), said instruction-execution-controlling portion performs register-to-**register** addressing processing for selecting another **register** (**register** at the address that R0 holds; in this, case **register at address 40h**).

> ***Note that the term register is broadly interpreted as any storage used by a computer (including RAM).***
>
> Because both references teach methods of addressing **memory locations (registers)**, it would have been obvious to substitute the direct register addressing, as taught by Patterson, for indirect addressing, as taught by Vault to achieve the predictable results of accessing data from computer memory via register-to-register addressing.
> [Italicized emphasis in original; Bold emphasis added.]

As best understood by Applicant, the stated rejection alleges that because a register and any other memory (e.g., RAM) are storage used by a computer, one skilled in the art would have considered the indirect register-to-*non-register memory* addressing as discussed in Vault as being applicable to any storage locations in a computer.

This alleged position is further elaborated in the Office Action's ***Response to Arguments*** section:

> Vault teaches addressing a memory location (40h in the current rejection) based on a value held in another **memory location (R0)**. Firstly, the term register is extremely broad and encompasses any memory location in a processor. Secondly, even if Vault does not teach the exact memory configuration as Patterson, it does not render the combination unobvious. Merely because a reference discusses addressing a certain type of memory in a particular fashion does not mean that it would not have been obvious to use the same addressing in another type of memory.
> [Emphasis added.]

Applicant respectfully submits that such a position as alleged in the Office Action mischaracterizes the teachings of Vault, as Vault would be understood by those skilled in the art. While Applicant acknowledges that the terms register and memory each refer to storage locations in a computer, Applicant respectfully notes that in the 8051 instruction set architecture discussed by Vault, CPU *registers* (e.g., registers R0-R7) are clearly distinguished from other

memory storage locations (e.g., other memory locations in Internal RAM, which RAM may, in some implementations, also be the same physical and/or logical storage unit in which the CPU registers are implemented) at least inasmuch as data in the other memory storage locations cannot be operated on by the ALU without first moving the data from the other memory storage location to a CPU register. The indirect addressing instruction discussed by Vault, and relied on in the Office Action, is one way of moving data from the other memory locations (where the ALU cannot operate directly on the data therein) to a register (e.g., so the data can then be operated on by the ALU pursuant to another instruction). Vault also explains that such indirect addressing is used for accessing memory locations above the first 128 bytes of Internal RAM, moving data from those locations to registers where the data can be operated on by the ALU.

In other words, in the 8051 instruction set architecture discussed inVault, a CPU *register* storage location is distinguished from another memory location based on their respective physical and/or logical relationship with respect to the ALU, and the indirect addressing instruction, as discussed above, has a particular use and function particular to, and inextricably related to, such registers *and* such other memory locations. As such, Applicant respectfully submits that those skilled in the art would not have viewed a *register* storage location in the context of the 8051 architecture as discussed by Vault as being similar to other memory locations that hold data that must be transferred to another storage location (e.g., to a register) for operation by an ALU, and thus would not have somehow considered the indirect addressing instruction as discussed in Vault as somehow implying, suggesting, or being applicable to, addressing between/among storage locations of a storage array (e.g., register array) that provides data from the storage locations to an ALU input without further addressing an intervening register.

Accordingly, Applicant respectfully submits that claim 1 is patentable over Vault and Patterson because these references, individually or in combination, do not disclose or suggest the combination of limitations recited in claim 1 (e.g., including claim 1 limitations directed to register-to-register addressing processing within a register array that is coupled to an operation portion input, as discussed above), which has been further clarified by the present amendment.

Additionally, Applicant respectfully submits that the § 103 rejection of claim 1 cannot stand because the Office Action's alleged rationale for concluding obviousness is improper and incorrect. First, Applicant submits that, contrary to the "obvious to substitute" rationale alleged in the Office Action, the combination of limitations recited in claim 1 cannot be met by *substituting* the indirect addressing of Vault for the direct addressing of Patterson because, for the reasons discussed above, those skilled in the art would not have considered Vault's indirect addressing as being applicable to register-to-register addressing in a register array.

Second, Applicant respectfully submits that it would not have been obvious to substitute, or in some way modify, the addressing modes of Patterson's MIPS architecture in a way that would add clock cycles per instruction and/or add additional instructions because such substitutions or modifications are contrary to the fundamental design philosophy of Patterson's MIPS architecture—a *reduced instruction set* computer (RISC) architecture. Thus, even assuming *arguendo* that the teachings of Vault would have been considered applicable to register-to-register addressing processing, Applicant submits that one skilled in the art would not have somehow *substituted* register-to-register addressing processing for Patterson's direct addressing because doing so would increase the number of clock cycles per instruction.

Similarly, even assuming *arguendo* that the teachings of Vault would have been considered applicable to register-to-register addressing processing, Applicant submits that one skilled in the art would not have otherwise *added* a register-to-register addressing processing mode to Patterson because doing so would increase the number instructions.

Further, Applicant's herein amendment expressly requiring both "a first access method" directed to register-to-register addressing processing and "a second access method" relating to direct addressing of the register array further supports the non-obviousness over the combination of Patterson and Vault at least inasmuch as it cannot be said that both of the recited access method limitations would be literally met if register-to-register processing would somehow be provided as a *substitute* for direct addressing in Patterson's register array.

Moreover, Applicant submits that the non-obviousness of Applicant's claimed invention (claim 1) is further evidenced by the long-felt needs it addresses. For example, as explained throughout the specification, Applicant's claimed invention (claim 1) provides for conserving or reducing chip real estate while increasing processing performance (e.g., high speed operation processing), which, Applicant submits, have been long-standing and ever-present goals driving the advancement of processors. Given such long-felt needs for improving processors, and given that processer architectures having a register file coupled to an ALU such as the MIPS architecture discussed in Patterson, and indirect addressing techniques such as the indirect addressing discussed in Vault, were also known well in advance of Applicant's claimed invention, Applicant's claimed invention cannot be said to be an obvious combination of the teachings of Patterson and Vault, nor a predictable use of prior art elements according to their established functions. Applicant respectfully submits that the combination of features recited in claim 1 also cannot be said to represent a predictable use of prior art elements (e.g., elements

discussed in Patterson and Vault) according to their established functions at least inasmuch as there are myriad possible approaches (increasing clock speed, reducing clock cycles per instruction, executing instructions concurrently, scaling devices, etc.) for addressing such long-felt needs, each such approach itself having myriad possible ways being achieved or implemented.

Therefore, for at least the foregoing reasons, Applicant respectfully submits that claim 1 is patentable.

Claim 14 recites, *inter alia*:

> An operation-processing method for performing operation processing based on an arbitrary operation program, said method comprising:
>
> accessing ***a register array*** having a plurality of registers, each for holding an arbitrary value based on a write address and a write control signal and for ***outputting the held value to an input of an operation portion based on a read address without further addressing a registers***;
>
> decoding an operation instruction from said operation program, said operation instruction including at leat one bit indicative of an access method;
>
> ***selecting one of said registers*** based on said operation instruction;
>
> ***in the event that said at least one bit is indicative of a first access method, (i) performing register-to-register addressing processing for selecting, based on a value held by said selected register, another of said registers of said register array, and (ii) performing with said operation portion an operation on a value held by said selected another register; and***
>
> ***in the event that said at least one bit is indicative of a second access method, performing with said operation portion an***

operation on a value held by said selected register, and not performing register-to-register addressing processing for selecting, based on a value held by said selected register, another of said registers of said register array.

Based on reasoning similar to that presented above with respect to claim 1, Applicant respectfully submits that the above combination of features recited in independent claim 14 are neither taught nor suggested by Patterson and Vault, individually or in combination, and thus claim 14 is also patentable.

Each of the other claims in this application is dependent on an independent claim discussed above, and is therefore believed patentable for at least the same reasons presented for the independent claim upon which it depends. Since each dependent claim is also deemed to define an additional aspect of the invention, however, the individual reconsideration of the patentability of each on its own merits is respectfully requested.

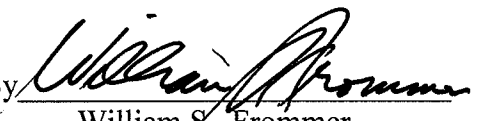
CONCLUSION

In view of the above, it is submitted that all pending claims are patentable and the application is in condition for allowance, and Applicant respectfully requests early reconsideration and allowance of the application.

Applicant gratefully acknowledges the Examiner's consideration of this matter, and the Examiner is respectfully invited to contact Applicant's undersigned representative by telephone on any outstanding issue regarding the application.

Respectfully submitted,

FROMMER LAWRENCE & HAUG LLP
Attorneys for Applicants

By 

William S. Frommer
Reg. No. 25,506
(212) 588-0800